https://ntrs.nasa.gov/search.jsp?R=19950025781 2020-06-16T06:12:24+00:00Z

NASA-CR-199034

IN-39-CR
P-32

CU-CSSC-90-07

CENTER FOR SPACE STRUCTURES AND CONTROLS

STAGGERED SOLUTION PROCEDURES FOR MULTIBODY DYNAMICS SIMULATIONS

(NASA-CR-199034) STAGGERED SOLUTION PROCEDURES FOR MULTIBODY DYNAMICS SIMULATION (Colorado Univ.) 32 p

N95-32202

Unclas

G3/39 0060384

by

K. C. Park, J. C. Chiou and J. D. Downer

April 1990

COLLEGE OF ENGINEERING
UNIVERSITY OF COLORADO
CAMPUS BOX 429
BOULDER, COLORADO 80309

STAGGERED SOLUTION PROCEDURES FOR MULTIBODY DYNAMICS SIMULATION

K. C. Park, J. C. Chiou and J. D. Downer

Department of Aerospace Engineering Sciences and
Center for Space Structures and Controls
University of Colorado
Boulder, Colorado 80309-0429

April 1990

Report No. CU-CSSC-90-07

Staggered Solution Procedures for Multibody Dynamics Simulation

K. C. Park, J. C. Chiou, and J. D. Downer
Department of Aerospace Engineering Sciences
and Center for Space Structures and Controls
University of Colorado at Boulder
Boulder, CO 80309-0429, USA

I. Introduction

Simulation of multibody dynamics systems – such as robotic manipulators, automobiles maneuvering and satellites deployment – remains a challenge to the dynamist due to its increasing roles in design improvements, control and safe operation. Because of substantial progress made during the past three decades in formulation[1-19], constraint treatment and solution techniques[21-36] and the availability of multibody dynamics simulation packages[37-42], it has now become almost a routine practice to perform realistic modeling and assessment of some practical problems such as mechanical linkages and manipulations of robotic arms if multibody components consist mostly of rigid bodies, discrete springs and dampers (see, e.g., Haug[15]). However, substantial advances in modeling, formulation and computational methods are necessary in order to develop a real-time simulation capability for ground vehicle maneuvering dynamics, robotic manipulations and space structures deployment/assembly.

Specifically, improved modeling of flexibility for localized motions and geometric nonlinearities, material nonlinearities and contact/friction phenomena, robust and accurate treatment of the system constraint conditions and efficient use of emerging computer hardware/software technology continue to offer intense research opportunities. Thus, the development of a real-time multibody dynamics simulation capability requires a concerted integration of various modeling, formulation and computational aspects. These include: selection of a data structure for describing the system topology, computerized generation of the governing equations of motion, implementation of suitable solution algorithms, incorporation of constraint conditions and easy interpretation of the simulation results. Of these, this chapter is concerned with three computational aspects of multibody dynamics simulation: direct time integration of the governing equations of motion, stabilization of constraint solution process and their computer implementation aspects.

From the computational viewpoint, multibody dynamics (MBD) problems are distinct from the structural dynamics problems in that the solution of MBD problems must also

satisfy, at each time integration step, the attendant kinematic and equilibrium constraints. This has motivated many dynamists to develop various techniques, in addition to direct integration algorithms, for accurately and efficiently handling the system constraints. Hence, reliability and cost of existing MBD simulation packages have been strongly affected by how efficiently and accurately the constraints are preserved during the numerical solution stage.

In general, there have been two types of direct time integration algorithms for the transient response analysis of dynamical systems: explicit and implicit algorithms (see, e.g., Hughes and Belytschko[43], Park[44] and Belytschko, Englemann and Liu[45]). Currently, implicit algorithms appear to be favored by many MBD specialists when both the generalized coordinates and the constraint forces are treated as the unknowns. In this case, the corresponding formulations incorporate the system constraints by the Lagrange multipliers method. It has been well known that the resulting Newton-like solution matrix is stiff. This has led to implicit time discretization of the constraint-augmented equations and simultaneous solution of both the generalized coordinates and the Lagrange multipliers. This approach has been extensively investigated by Gear[21], Baumgarte[22,29] Orlandea, Chase and Calahan[23], Petzold[27], Nikravesh[31], among others. Because these methods solve both the generalized coordinates and the constraint forces simultaneously, they will be called the *simultaneous solution methods* in this chapter.

On the other hand, if the constraints are eliminated so as to reduce the number of unknowns, it is possible for one to employ either implicit or explicit algorithm. For this situation, one may invoke either a geometric or algebraic procedure to streamline the resulting equations of motion if the system topology is an open tree. In essence, geometric procedures have utilized an open-tree topology such as the use of the incidence matrix by Wittenburg[10] and the body array matrix by Huston[19]. Some of the proposed algebraic procedures include the singular decomposition by Walton et al[20], the use of the generalized speed of Kane and Levinson[20], the coordinate partitioning technique by Wehage and Haug[28], the selection of independent coordinates through the natural-coordinate formulation of Garcia de Jalon et al[33] and the so-called order-N procedures of Armstrong[11], Hollerbach[12], Schwertassek and Roberson[17], Orin, et al[25], among others.

As the complexity of MBD systems increases, the simultaneous solution methods have become less attractive. This is due to matrix ill-conditioning especially for the so-called *index* two and higher index problems (see, e.g., Ref. 27 and Brenan, Campbell and Petzold[46] for the definition of *index* for constraint characterization), divergence of the solution away from the constraint conditions, and ultimately, due to a large size of the equations that must be handled. As an alternative to the simultaneous solution methods, a series of computational methods that employ a *divide-and-conquer* strategy have been developed, which are termed as *partitioned solution procedures* presented in Park[47], Felippa and Park[48] and Park and Felippa[49]. As an example, partitioned solution procedures allow one to analyze fluid-structure interaction problems with two separate single-field analysis packages, namely, the structural dynamics module and the fluid dynamics analyzer. At each time integration step, one may advance the solution of structural equations of motion by treating the fluid coupling term as an external force. Once the structural coordinates are advanced, the fluid state variables can be advanced by treating the structural coupling

terms as a source term. A naive partitioned procedure, however, can suffer from a loss of accuracy as well as computational stability. Thus, a combination of equation augmentation and stabilization should be devised to recover the accuracy loss and maintain unconditional stability. Such a solution procedure is in contrast to a practice of embedding both the structural and fluid dynamics attributes into a combined analysis program.

The numerical solution procedure for MBD systems which we advocate in this chapter is termed a *staggered MBD solution porcedure* that solves the generalized coordinates in a separate module from that for the constraint force. This requires a reformulation of the constraint conditions so that the constraint forces can also be integrated in time. A major advantage of such a partitioned solution procedure is that additional analysis capabilities such as active controller and design optimization modules can be easily interfaced without embedding them into a monolithic program. To this end, the rest of the chapter is organized as follows.

After introducing the basic equations of motion for MBD system in the next section, Section III briefly reviews some constraint handling techniques and introduces the staggered stabilized technique[34,35] for the solution of the constraint forces as independent variables.

The numerical direct time integration of the equations of motion is described in Section IV. As accurate damping treatment is important for the dynamics of space structures, we have employed the central difference method and the mid-point form of the trapezoidal rule since they engender no numerical damping. This is in contrast to the current practice in dynamic simulations of ground vehicles by employing a set of backward difference formulas[46]. First, the equations of motion is partitioned according to the translational and the rotational coordinates. This sets the stage for an efficient treatment of the rotational motions via the singularity-free Euler parameters. The resulting partitioned equations of motion are then integrated via a two-stage explicit stabilized algorithm for updating both the translational coordinates and angular velocities[34]. Once the angular velocities are obtained, the angular orientations are updated via the mid-point implicit formula employing the Euler parameters.

When the two algorithms, namely, the two-stage explicit algorithm for the generalized coordinates and the implicit staggered procedure for the constraint Lagrange multipliers, are brought together in a staggered manner, they constitute a staggered explicit-implicit procedure which are summarized in Section V. Section VI presents some example problems and discussions concerning several salient features of the staggered MBD solution procedure are offered in Section VII.

II. Governing Equations of Motion

The Lagrangian equations of motion for mechanical systems that are free from any constraint can be written, for the generalized coordinate component u_i, as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{u}_i} - \frac{\partial L}{\partial u_i} = Q_i, \quad i = 1 \ldots n. \tag{1}$$

where L is the system Lagrangian, t is the time, $(\dot{})$ denotes time differentiation and Q_i is the generalized applied force. It is well-known that, if there are m-constraint conditions

imposed on $\{u_i, \quad i = 1 \ldots n\}$, the above equation must be modified as

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{u}_i} - \frac{\partial L}{\partial u_i} = Q_i + \sum_{k=1}^{m} \lambda_k B_{ki}, \quad i = 1 \ldots n, \tag{2}$$

where λ is the Lagrange multiplier and B_{ki} is the i-th gradient component of the k-th constraint equation, *viz*, for configuration constraints

$$\Phi_k(\mathbf{u}) = 0, \quad B_{ki} = \frac{\partial \Phi_k}{\partial u_i}, \quad k = 1 \ldots m \tag{3}$$

and for motion constraints

$$\Phi_k(\mathbf{u}, \dot{\mathbf{u}}) = 0, \quad B_{ki} = \frac{\partial \Phi_k}{\partial \dot{u}_i}, \quad k = 1 \ldots m. \tag{4}$$

Therefore, regardless of the nature of constraints one may express the equations of motion with constraints in the following form:

$$\begin{bmatrix} M & B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} \ddot{\mathbf{u}} \\ \lambda \end{pmatrix} = \begin{pmatrix} Q \\ c \end{pmatrix} \tag{5}$$

where M is a positive-definite matrix and c depends on the nature of constraints. For example, for configuration constraints we have

$$c = -\frac{\partial}{\partial u}\left(\frac{\partial \Phi}{\partial u}\dot{u}\right) - 2\frac{\partial}{\partial t}\left(\frac{\partial \Phi}{\partial u}\dot{u}\right) - \frac{\partial^2 \Phi}{\partial t^2} \tag{6}$$

and for motion constraints

$$c = -\frac{\partial \Phi}{\partial t}. \tag{7}$$

An implicit time integration formula to solve (5) may be written as

$$\begin{cases} \dot{\mathbf{u}}^n = \delta \ddot{\mathbf{u}}^n + \mathbf{h}_{\dot{u}}^n \\ \mathbf{u}^n = \delta \dot{\mathbf{u}}^n + \mathbf{h}_u^n \end{cases} \tag{8}$$

where δ is a stepsize that is dependent on the choice of formula, and $\mathbf{h}_{\dot{u}}^n$ and $\mathbf{h}_u^n$ are formula-dependent historical vectors that consist of past-step solution components[48,50]. As an example, the trapezoidal rule has the following δ and historical vectors

$$\begin{cases} \delta = h/2 \\ \mathbf{h}_{\dot{u}}^n = \dot{\mathbf{u}}^{n-1} + \delta \ddot{\mathbf{u}}^{n-1} \\ \mathbf{h}_u^n = \mathbf{u}^{n-1} + \delta \dot{\mathbf{u}}^{n-1} \end{cases} \tag{9}$$

where h is the time-step increment.

Substitution of (8) into (5) yields

$$\begin{bmatrix} M & \delta^2 B^T \\ B & 0 \end{bmatrix} \begin{pmatrix} \mathbf{u}^n \\ \lambda^n \end{pmatrix} = \begin{pmatrix} r_u^n \\ r_\lambda^n \end{pmatrix} = \begin{pmatrix} Mh^n + \delta^2 Q^n \\ Bh^n + \delta^2 c \end{pmatrix} \tag{10}$$

In practice, in order to avoid pivoting and to maintain high accuracy, the solution of the above difference equations is carried out as follows. First, since M is nonsingular for properly formulated dynamical problems, one computes

$$\mathbf{u}_u = M^{-1}\mathbf{r}_u^n, \quad C = M^{-1}B^T, \quad A = BC \tag{11}$$

and factors A. Second, one obtains λ^n by solving

$$\lambda^n = A^{-1}(B\mathbf{u}_u - \mathbf{r}_\lambda^n)/\delta^2 \tag{12}$$

Finally, $\mathbf{u}^n$ is obtained from

$$\mathbf{u}^n = \mathbf{u}_u - \delta^2 C\lambda^n \tag{13}$$

It should be noted that the accuracy loss associated with the factoring of an ill-conditioned matrix $BA^{-1}B^T$ and the subsequent backsubstitutions can severely influence the solution accuracy of not only the Lagrange multipliers but also the generalized coordinates as seen from (12) and (13). This has motivated many numerical analysts to undertake the development of methods for differential-algebraic systems as the recent monograph[46] and references therein attest to their rich numerical properties. It is generally agreed that the present status of differential-algebraic methods yield robust solutions for problems of *index one*, but can suffer from inaccurate solutions of the Lagrange multipliers for higher *index* problems. Observe that many practical multibody dynamics problems are characterized by index greater than one. Hence, the need to compute accurately the constraint forces remains a challenge. For instance, for lock-up mechanisms that are activated when truss structures are fully deployed in space often introduce stiff responses with nearly singular state of $BM^{-1}B^T$. It is with these problems for which more robust constraint computation algorithms are called for.

One way to improve the accuracy of constraint force computations is to adopt index reduction strategies as discussed in Ref. 46. However, index reduction inevitably introduces additional system degrees of freedom in the resulting differential-algebraic equations, thus destroying the matrix sparsity of (5) in addition to the increased size of the matrix B. In what follows we present an alternative approach based on a parabolic regularization of the equations for the Lagrange multipliers, which preserves the first row of (5) and enables us to solve λ from the parabolic differential equations.

III. Constraint Handling Techniques

As alluded to in Introduction, techniques for handling the system constraints constitute a major part of solution procedures for the numerical simulation of multibody dynamics systems. In this section, we will first review the coordinate partitioning technique, Baumgarte's technique and the penalty technique. The staggered stabilization procedure

which we advocate will then be described in detail. A distinct feature of the staggered stabilization procedure is that it can be implemented in a stand-alone module, thus can be interfaced not only with the equation solver for rigid-body systems but with that for flexible-body systems as well.

A. Coordinate Partitioning Technique

In the coordinate partitioning[28,33] or singular decomposition technique[20,30], one selects a rank sufficient part of **B** and partitions it as

$$B = \lfloor B_i \quad B_e \rfloor, \quad \mathbf{u} = \lfloor \mathbf{u}_i \quad \mathbf{q}_e \rfloor \tag{14}$$

where the rank of $B_i(m \times m)$ is m and the subscripts (i, e) refer to *internal* and *external* variables, respectively. First, we express $\mathbf{u}_i$ in terms of $\mathbf{u}_e$ as

$$\mathbf{u}_i^n = B_i^{-1}(\mathbf{r}_\lambda^n - B_e \mathbf{u}_e^n) \tag{15}$$

Since we have

$$\lfloor -B_e^T B_i^{-T} \quad I \rfloor \begin{Bmatrix} B_i^T \\ B_e^T \end{Bmatrix} = 0 \tag{16}$$

The first row of (10) reduces to

$$(M_e + T^T M_i T)\mathbf{q}_e^n = \mathbf{r}_e^n \tag{17}$$

where

$$T = B_i^{-1} B_e, \quad M = \begin{bmatrix} M_i & 0 \\ 0 & M_e \end{bmatrix}, \quad \mathbf{r}_u^n = \begin{Bmatrix} \mathbf{r}_{u_i}^n \\ \mathbf{r}_{u_e}^n \end{Bmatrix} \tag{18}$$

and

$$\mathbf{r}_e^n = \mathbf{r}_{u_e}^n - T^T \mathbf{r}_{u_i}^n + T^T M_i B_i^{-1} \mathbf{r}_\lambda^n \tag{19}$$

Once one obtains $\mathbf{u}_e^n$, one can obtain $\mathbf{u}_e^n$ from (15) and similarly λ from (12). Note that even though (17) has a smaller dimension than that of (10a), its left-hand side matrix is in general full since T given by (18a) is in general full. Hence, unless T is a constant matrix, one must refactor the solution matrix in (17) whenever a new T is formed.

B. Baumgarte's Technique

Baumgarte's technique[22,29] is based on the observation that the errors committed in computing the constraint conditions (3) or (4) can either be critically damped out or exponentially decreased as the integration process continues. Mathematically, this can be stated for the configuration constraint equation(3) as

$$\ddot{\Phi} + 2\alpha\dot{\Phi} + \beta\Phi = 0 \tag{20}$$

or the motion constraint equation(4) as

$$\dot{\Phi} + \gamma\Phi = 0 \tag{21}$$

In terms of the general constraint equation augmentation as given by (5b), the preceding stabilization is equivalent to modifying c in (5b) accordingly. Hence, the technique can be implemented within the standard augmented form of the equations of motion (5). However, if $BM^{-1}B^T$ is ill-conditioned, which can happen since B is in general state-dependent, the accuracy of generalized constraint force, λ, can be considerably degraded. This can occur if any two rows of B are physically similar (i.e., when two members form a straight line) or numerically close during three-dimensional orientations.

C. Penalty Technique

In the two constraint handling techniques outlined so far, the objective was to satisfy the constraint condition

$$\Phi = 0 \tag{22}$$

whose differentiated forms were augmented to the equations of motion. In the penalty procedure, one adopts

$$\lambda = \frac{1}{\epsilon}\Phi, \quad \epsilon \to 0 \tag{23}$$

as the basic constraint equations instead of the twice-differentiated form adopted in (5).

It is noted that the penalty formulation tacitly assumes that there will be violations of the constraint condition in actual computations as discussed in Lanczos[51]. If one substitutes (23) into the governing equations of motion, the resulting equation becomes

$$M\ddot{\mathbf{u}} + \frac{1}{\epsilon}B^T\Phi = Q \tag{24}$$

A major drawback of the above penalty procedure is that, once an error is committed in computing λ, there is no compensation scheme by which the drifting of the numerical solution can be corrected. This has led to the development of a staggered stabilized procedure as described below.

D. Staggered Stabilization Procedure

To illustrate this procedure we will consider the case of nonholonomic constraints. Instead of substituting the penalty expression directly into the governing equations of motion, first we differentiate (23) once to obtain

$$\dot{\boldsymbol{\lambda}} = \frac{1}{\epsilon}(\mathbf{B}\ddot{\mathbf{u}} + \frac{\partial\Phi}{\partial t}) \tag{25}$$

where we assume the penalty parameter, ϵ, to be constant.

Second, we obtain for $\ddot{\mathbf{u}}$ from (5a) in the form

$$\ddot{\mathbf{u}} = \mathbf{M}^{-1}(\bar{\mathbf{Q}} - \mathbf{B}^T\boldsymbol{\lambda}) \tag{26}$$

and substitute it into (25) to yield

$$\epsilon\dot{\boldsymbol{\lambda}} + \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T\boldsymbol{\lambda} = \mathbf{r}_\lambda = \mathbf{B}\mathbf{M}^{-1}\bar{\mathbf{Q}} + \frac{\partial\Phi}{\partial t} \tag{27}$$

Notice that the homogeneous part of the above stabilized equation in terms of the generalized constraint forces, $\boldsymbol{\lambda}$, has the following companion eigenvalue problem:

$$(\gamma + \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T/\epsilon)\mathbf{y} = 0 \tag{28}$$

where $\{\gamma_k, \quad k = 1 \ldots m\}$ are the eigenvalues of the homogeneous operator for the new stabilized constraint equations (27). Since γ_k also dictates how the errors in the constraint forces will diminish with time, the errors committed in the constraint conditions will decay with their corresponding different response time constants. This physically oriented stabilization property of the present technique is in contrast to that of Baumgarte's technique wherein all the error components diminish according to a single time constant.

Third, this technique enables one to solve for $\boldsymbol{\lambda}$ from the stabilized differential equation (27). Specifically, one now has two coupled equations, one set for the generalized coordinates $\mathbf{u}$ and the other for the generalized constraint forces $\boldsymbol{\lambda}$, which are recalled here from (5a) and (27) for the case of nonholonomic constraints:

$$\begin{bmatrix} \mathbf{M} & 0 \\ 0 & \epsilon \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{u}} \\ \dot{\boldsymbol{\lambda}} \end{Bmatrix} + \begin{bmatrix} 0 & \mathbf{B}^T \\ 0 & \mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{u}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \bar{\mathbf{Q}} \\ \mathbf{r}_\lambda \end{Bmatrix} \tag{29}$$

Note that the above coupled equations directly provide the desired differential equations for a pair of $\lfloor \mathbf{u} \quad \boldsymbol{\lambda} \rfloor$.

For holonomic constraints, one has several stabilization possibilities. The one we have chosen is to integrate the governing equations of motion once to obtain

$$\dot{\mathbf{u}}^n = \delta\mathbf{M}^{-1}(\bar{\mathbf{Q}}^n - \mathbf{B}^T\boldsymbol{\lambda}^n) + \mathbf{h}_{\dot{u}}^n \tag{30}$$

which is substituted into

$$\dot{\boldsymbol{\lambda}} = \frac{1}{\epsilon}(\mathbf{B}\dot{\mathbf{u}} + \frac{\partial \Phi}{\partial t}) \tag{31}$$

to yield:

$$\epsilon\dot{\boldsymbol{\lambda}}^n + \delta\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T\boldsymbol{\lambda}^n = \mathbf{B}(\delta\mathbf{M}^{-1}\bar{\mathbf{Q}}^n + \mathbf{h}_{\dot{u}}^n) + \frac{\partial \Phi}{\partial t} \tag{32}$$

It is observed that, even if $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T$ is almost singular, this stabilization technique as derived in (27) and (32) would not cause numerical difficulty in computing $\boldsymbol{\lambda}$ since the solution iteration matrix becomes $(\epsilon + \delta\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T)$ for nonholonomic cases and $(\epsilon + \delta^2\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T)$ for holonomic cases. It is noted that one must choose ϵ in such a way to maintain robust solution when $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T$ becomes ill-conditioned by choosing $\epsilon \sim c/|(\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T)^{-1}| \cdot |\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T|$ where c is the solution accuracy desired for λ.

Integration of the above equation by the mid-point implicit rule yields the following difference equation:

$$\begin{cases} (\epsilon\mathbf{I} + \frac{h}{4}\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T)\boldsymbol{\lambda}^{n+1/4} = \frac{h}{4}(\mathbf{r}_\lambda^{n+\frac{1}{2}} + \mathbf{r}_\lambda^n) + \epsilon\boldsymbol{\lambda}^n \\ \boldsymbol{\lambda}^{n+1/2} = 2\boldsymbol{\lambda}^{n+1/4} - \boldsymbol{\lambda}^n \end{cases} \tag{33}$$

It has been shown that the staggered stabilized procedure for the solution of the constraints offers not only a modular software package to treat the constraints but also has been found to yield more robust solutions compared to the techniques proposed by Baumgarte as reported in Park and Chiou[35]. In particular, even when $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T$ becomes nearly singular, the staggered stabilized procedure (33) gives stable and acceptable solutions whereas the constraint forces computed by the Baumgarte's technique diverge.

IV. Solution Algorithms for Generalized Coordinates

In addition to the choice of implicit and explicit formulas, the recognition that the equations of motion for multibody systems with constraints are not ordinary differential equations (ODEs) (see, e.g., Petzold[27]) has placed a unique requirement in the selection of solution algorithms for multibody dynamics problems. From the user's viewpoint, one has the option of either employing one of the available ODE packages (see Enright[32] for existing ODE packages) or building a special solution module. It should be noted that, since the integration of angular velocity vector does not lead to angular orientations, one must solve a set of kinematical equations to obtain the desired angular orientations.

In this section we describe an explicit-implicit transient analysis algorithm that exploits the special kinematical relationships of the generalized rotational coordinates vs. the angular velocity, namely, the Euler parameters[34]. The integration of the translational coordinates and the angular velocity is accomplished by the central difference formula. It should be mentioned that the use of the central difference formula does impose a stepsize restriction due to its stability limit ($\omega_{max}h \leq 2$) where ω_{max} is the highest angular velocity of the system components for rigid-body systems or the highest frequency of the entire flexible members for flexible-body systems. The simplicity of its programming effort and robustness of its solution results can often become compelling enough to adopt an explicit formula, which is the view taken here.

In conventional structural dynamics analysis, explicit time integration of the equations of motion by the central difference formula involves the following two updates per step:

$$\begin{cases} \dot{\mathbf{u}}^{n+1/2} = \dot{\mathbf{u}}^{n-1/2} + h\ddot{\mathbf{u}}^n \\ \mathbf{u}^{n+1} = \mathbf{u}^n + h\dot{\mathbf{u}}^{n+1/2} \end{cases} \tag{34}$$

Unfortunately, this simplistic procedure is not directly applicable to the rotational part of the equations of motion as $\boldsymbol{\omega}$ is not directly integrable, except for some special kinematic configurations. This motivates us to partition $\dot{\mathbf{q}}$ into the translational velocity vector, $\dot{\mathbf{d}}$, which is directly integrable and the angular velocity vector, $\boldsymbol{\omega}$, which is not, and treat them differently, viz.:

$$\ddot{\mathbf{u}} = \begin{Bmatrix} \ddot{\mathbf{d}} \\ \dot{\boldsymbol{\omega}} \end{Bmatrix}, \qquad \dot{\mathbf{u}} = \begin{Bmatrix} \dot{\mathbf{d}} \\ \boldsymbol{\omega} \end{Bmatrix} \tag{35}$$

The equations of motion (5a) can be partitioned according to the above partitioning:

$$\begin{bmatrix} M_d & 0 \\ 0 & M_\omega \end{bmatrix} \begin{Bmatrix} \ddot{\mathbf{d}} \\ \dot{\boldsymbol{\omega}} \end{Bmatrix} = \begin{Bmatrix} Q_d \\ Q_\omega \end{Bmatrix} \tag{36}$$

where

$$\begin{Bmatrix} Q_d \\ Q_\omega \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_d - \mathbf{D}_d(\dot{\mathbf{d}}) - \mathbf{S}_d(\mathbf{d},\mathbf{e}) - \mathbf{B}_d^T\boldsymbol{\lambda} \\ \mathbf{f}_\omega - \mathbf{D}_\omega(\boldsymbol{\omega}) - \mathbf{S}_\omega(\mathbf{d},\mathbf{e}) - \mathbf{B}_\omega^T\boldsymbol{\lambda} \end{Bmatrix} \tag{37}$$

in which the subscripts (d,ω) refer to the translational and the rotational motions, respectively, $\mathbf{f}$ is the external force vector, $\mathbf{D}$ is the generalized damping force including the centrifugal force, $\mathbf{S}$ is the internal force vector including member flexibility, $\mathbf{q}$ is the angular orientation parameters, $\mathbf{B_d}$ and $\mathbf{B}\boldsymbol{\omega}$ are the partition of the combined gradient matrices of the constraint conditions (3) or (4) that are symbolically expressed as

$$\mathbf{B} = \mathbf{B}_N + \mathbf{B}_H, \qquad \boldsymbol{\lambda} = \boldsymbol{\lambda}_N + \boldsymbol{\lambda}_H \tag{38}$$

To effect the body-by-body integration for the rotational degrees of freedom, we partition $\dot{\boldsymbol{\omega}}$ further into

$$\dot{\boldsymbol{\omega}} = \lfloor \dot{\boldsymbol{\omega}}^1, \dot{\boldsymbol{\omega}}^2, \ldots, \dot{\boldsymbol{\omega}}^P \rfloor^T \tag{39}$$

where $\dot{\boldsymbol{\omega}}^{(j)}$ is a (3×1) angular acceleration vector for the j-th body,

$$\dot{\boldsymbol{\omega}}^{(j)} = \lfloor \omega_1^{(j)}, \omega_2^{(j)}, \omega_3^{(j)} \rfloor \tag{40}$$

We now present the update algorithm for both translational and rotational coordinates.

A. Update of Translational and Angular Velocity

First, assume that $\mathbf{d}^{n+1/2}$ and $\mathbf{q}^{n+1/2}$ are already computed so that we can compute $\ddot{\mathbf{d}}^{n+1/2}$ and $\dot{\boldsymbol{\omega}}^{n+1/2}$ by (36), namely,

$$\begin{Bmatrix} \ddot{\mathbf{d}}^{n+1/2} \\ \dot{\boldsymbol{\omega}}^{n+1/2} \end{Bmatrix} = -M^{-1} \begin{Bmatrix} \mathbf{D}_d^{n+\frac{1}{2}} + \mathbf{S}_d^{n+\frac{1}{2}} - \mathbf{B}_d^T\boldsymbol{\lambda}^{n+\frac{1}{2}} \\ \mathbf{D}_\omega^{n+\frac{1}{2}} + \mathbf{S}_\omega^{n+\frac{1}{2}} - \mathbf{B}_\omega^T\boldsymbol{\lambda}^{n+\frac{1}{2}} \end{Bmatrix} \tag{41}$$

Second, we update the translational velocity and the angular velocity vectors at the step (n+1) by

$$\begin{cases} \dot{\mathbf{d}}^{n+1} = \dot{\mathbf{d}}^n + h\ddot{\mathbf{d}}^{n+1/2} \\ \boldsymbol{\omega}^{n+1} = \boldsymbol{\omega}^n + h\dot{\boldsymbol{\omega}}^{n+1/2} \end{cases} \tag{42}$$

Third, we update the translational displacement, $\mathbf{d}$, by

$$\mathbf{d}^{n+3/2} = \mathbf{d}^{n+1/2} + h\dot{\mathbf{d}}^{n+1} \tag{43}$$

However, the updating of the angular orientation requires somewhat involved computations. To this end, we will employ the Euler parameters and update them accordingly.

B. Update of Euler Parameters and Angular Velocity

As mentioned in conjunction with a direct use of (34) for integrating the rotational equations of motion, it is necessary for one to introduce a set of generalized coordinates whose time rate can be related to the angular velocity. To this end, we employ the four-parameter Euler representation of the angular velocity for each body as (see, e.g., Wittenburg[10]):

$$\dot{\mathbf{q}} = \frac{1}{2}\begin{bmatrix} 0 & -\boldsymbol{\omega}^T \\ \boldsymbol{\omega} & -\tilde{\boldsymbol{\omega}} \end{bmatrix} \mathbf{q} = A(\omega)\mathbf{q}, \quad \mathbf{q} = \lfloor q_0 \quad q_1 \quad q_2 \quad q_3 \rfloor^T \tag{44}$$

that is subject to the constraint:

$$\mathbf{q}^T\mathbf{q} = 1 \tag{45}$$

where

$$\tilde{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}, \qquad \boldsymbol{\omega} = \lfloor \omega_1 \quad \omega_2 \quad \omega_3 \rfloor^T \tag{46}$$

and the nodal-designation superscript is omitted for notational simplicity.

We adopt the mid-point implicit procedure to integrate the Euler parameters:

$$\begin{cases} \dot{\mathbf{q}}^{n+1} = \mathbf{A}(\boldsymbol{\omega}^{n+1}) \cdot \mathbf{q}^{n+1} \\ \mathbf{q}^{n+1} = \mathbf{q}^{n+1/2} + \frac{h}{2}\dot{\mathbf{q}}^{n+1} \\ \mathbf{q}^{n+3/2} = 2\mathbf{q}^{n+1} - \mathbf{q}^{n+1/2} \\ (\mathbf{q}^{n+3/2})^T \cdot \mathbf{q}^{n+3/2} = 1 \end{cases} \tag{47}$$

It should be noted that the mid-point implicit update is no more costly than any explicit as the solution matrix inversion can be explicitly obtained.

Finally, once $\mathbf{q}^{n+3/2}$ is computed from (47), it is often required to compute the body-fixed basis vector, $\mathbf{b} = \lfloor \mathbf{b}_1 \quad \mathbf{b}_2 \quad \mathbf{b}_3 \rfloor^T$ in terms of the inertial basis vectors, $\mathbf{e} = \lfloor \mathbf{e}_1 \quad \mathbf{e}_2 \quad \mathbf{e}_3 \rfloor^T$. These two vectors are related by

$$\mathbf{b} = \mathbf{R}\mathbf{e} \tag{48}$$

where

$$\mathbf{R} = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} \tag{49}$$

C. Update of $\dot{\mathbf{d}}, \omega, \mathbf{d}, \mathbf{q}$ at the $(n+2)$–step

So far we have advanced from the step (n+1) to the step (n+3/2). In other words, we have advanced only half of the total step. For the next step, viz, the step (n+2) from (n+3/2), we employ the following sequence of computations:

$$\begin{Bmatrix} \ddot{\mathbf{d}}^{n+1} \\ \dot{\omega}^{n+1} \end{Bmatrix} = -M^{-1} \begin{Bmatrix} \mathbf{D}_d^{n+1} + \mathbf{S}_d^{n+1} - \mathbf{B}_d^T \boldsymbol{\lambda}^{n+1} \\ \mathbf{D}_\omega^{n+1} + \mathbf{S}_\omega^{n+1} - \mathbf{B}_\omega^T \boldsymbol{\lambda}^{n+1} \end{Bmatrix} \tag{50}$$

$$\begin{cases} \dot{\mathbf{d}}^{n+3/2} = \dot{\mathbf{d}}^{n+1/2} + h\ddot{\mathbf{d}}^{n+1} \\ \omega^{n+3/2} = \omega^{n+1/2} + h\dot{\omega}^{n+1} \end{cases} \tag{51}$$

$$\begin{cases} \mathbf{d}^{n+2} = \mathbf{d}^{n+1} + h\dot{\mathbf{d}}^{n+3/2} \\ \dot{\mathbf{q}}^{n+3/2} = \mathbf{A}(\omega^{n+3/2})\mathbf{q}^{n+3/2} \\ \mathbf{q}^{n+3/2} = \mathbf{q}^{n+1} + \frac{h}{2}\dot{\mathbf{q}}^{n+3/2} \\ \mathbf{q}^{n+2} = 2\mathbf{q}^{n+3/2} - \mathbf{q}^{n+1} \\ (\mathbf{q}^{n+2})^T \mathbf{q}^{n+2} = 1 \end{cases} \tag{52}$$

Note that we do not use $\mathbf{d}^{n+3/2}$ and $\mathbf{q}^{n+3/2}$ in advancing from the step (n+3/2) to the present step (n+2) in computing $\mathbf{d}^{n+2}$ and $\mathbf{q}^{n+2}$. Instead, we employ $\mathbf{d}^{n+1}$ and $\mathbf{q}^{n+1}$, hence the name *two-stage staggered explicit procedure*[34]. The net result is that, even though we take a full step (h instead of h/2), we only advance half the step at a time. In other words, we evaluate the acceleration and the angular acceleration vectors twice for each full step.

V. Implementation

We will now outline the implementation aspects of the the partitioned MBD solution procedure. The procedure is implemented into two separate integration modules: generalized-coordinate integrator (CINT) and Lagrange multiplier solver (LINT). The generalized-coordinate integrator employs a two-stage modified form of the central difference method for updating the angular velocity vector and the mid-point implicit rule for updating the angular orientations via the Euler parameters. The Lagrange multipliers solver adopts a staggered form of the mid-point implicit method.

A. Generalized-Coordinate Integrator (CINT)

The module receives $f_\lambda^n = \mathbf{B}^T \boldsymbol{\lambda}^n$ from LINT and advances the solution of the MBD equation (1) from time $\mathbf{t}^n$ to $\mathbf{t}^{n+1}$. At each integration step, CINT performs the following computations.

Given: $\mathbf{p}^n = (\dot{\mathbf{d}}^{n-1/2},\ \mathbf{d}^n,\ \omega^{n-1/2},\ \mathbf{q}^n)$ and $\mathbf{g}^n = (\omega^n, f_\lambda^n = \mathbf{B}^T \boldsymbol{\lambda}^n))$

Compute: $\ddot{\mathbf{d}}^n$ and $\dot{\omega}^n$ by (41)

Advance:

$$\begin{cases} \dot{\mathbf{d}}^{n+1/2} = \dot{\mathbf{d}}^{n-1/2} + h\ddot{\mathbf{d}}^n \\ \mathbf{d}^{n+1} = \mathbf{d}^n + h\dot{\mathbf{d}}^{n+1/2} \end{cases} \tag{53}$$

$$\begin{cases} \boldsymbol{\omega}^{n+1/2} = \boldsymbol{\omega}^{n-1/2} + h\dot{\boldsymbol{\omega}}^{n} \\ \tilde{\mathbf{q}}^{n+1/2} = \frac{1}{\Delta}[\mathbf{I} + \frac{h}{2}\mathbf{A}(\boldsymbol{\omega}^{n+1/2})] \cdot \mathbf{q}^{n}, \quad \Delta = 1 + \frac{h^2}{4}(\omega_1^2 + \omega_2^2 + \omega_3^2) \\ \mathbf{q}^{n+1} = 2\tilde{\mathbf{q}}^{n+1/2} - \mathbf{q}^{n}, \quad (\mathbf{q}^{n+1})^T \cdot \mathbf{q}^{n+1} = 1 \end{cases} \tag{54}$$

Output: $\mathbf{p}^{n+1} = (\dot{\mathbf{d}}^{n+1/2},\ \mathbf{d}^{n+1},\ \boldsymbol{\omega}^{n+1/2}, \mathbf{q}^{n+1})$

Module Invoke: Call CINT $(\mathbf{p}^{n},\ \mathbf{g}^{n},\ h,\ \mathbf{p}^{n+1})$

where h is the stepsize and $\mathbf{A}(\boldsymbol{\omega})$ is given by

$$\mathbf{A}(\boldsymbol{\omega}) = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} \tag{55}$$

and $\tilde{\mathbf{q}}^{n+1/2}$ is an intermediate vector and (54c) must be solved to obtain $\mathbf{q}^{n+1}$ so as to to satisfy the linear dependency constraint, $\mathbf{q}^T\mathbf{q} = 1$.

B. Lagrange Multiplier Solver (LINT)

This module receives $(\dot{\mathbf{d}},\ \mathbf{d},\ \boldsymbol{\omega},\ \mathbf{q})$ from CINT and performs the following computations.

Given: $\ell^{n+1/2} = (\dot{\mathbf{d}}^{n+1/2},\ \mathbf{d}^{n+1/2},\ \boldsymbol{\omega}^{n+1/2},\ \mathbf{q}^{n+1/2},\ \boldsymbol{\lambda}^{n})$

Compute: $\mathbf{B}^{n+1/2}$, $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T$ and $\mathbf{r}_{\lambda}^{n+1/2}$ by (3) and (4)

Advance:

$$\begin{cases} \tilde{\boldsymbol{\lambda}}^{n+1/4} = (\epsilon\mathbf{I} + \frac{h}{4}\mathbf{B}M^{-1}B^T)^{-1}(\epsilon\boldsymbol{\lambda}^{n} + \frac{h}{4}(\mathbf{r}_{\lambda}^{n} + \mathbf{r}_{\lambda}^{n+1/2})) \\ \boldsymbol{\lambda}^{n+1/2} = 2\tilde{\boldsymbol{\lambda}}^{n+1/4} - \boldsymbol{\lambda}^{n} \\ \mathbf{f}_{\lambda}^{n+1/2} = (\mathbf{B}^{n+1/2})^T \cdot \boldsymbol{\lambda}^{n+1/2} \end{cases} \tag{56}$$

Output: $\boldsymbol{\lambda}^{n+1/2},\quad \mathbf{f}_{\lambda}^{n+1/2}$

Module Invoke: Call LINT $(\ell^{n+1/2},\ h,\ \boldsymbol{\lambda}^{n+1/2},\ \mathbf{f}_{\lambda}^{n+1/2})$

C. Two-Stage Explicit-Implicit Staggered Procedure

In order to evaluate $\dot{\boldsymbol{\omega}}^{n+1}$, $\boldsymbol{\omega}^{n+1}$ must be known. Notice from the preceding section that only $\dot{\boldsymbol{\omega}}^{n+1/2}$ is available. Because inaccurate treatments of the gyroscopic damping and the centrifugal force terms can lead quickly to computational instability in computing $\dot{\boldsymbol{\omega}}^{n+1}$, it is not advisable to obtain $\boldsymbol{\omega}^{n+1}$ by extrapolating with $\boldsymbol{\omega}^{n+1/2}$ and $\boldsymbol{\omega}^{n-1/2}$. To mitigate

this difficulty, we advance only to the next half step, at each CINT and LINT call. This is illustrated as follows:

$t = t^n$
Call CINT ($\mathbf{p}^n$, $\mathbf{g}^n$, h, $\mathbf{p}^{n+1}$)
Call LINT ($\ell^{n+1/2}$, h, $\boldsymbol{\lambda}^{n+1/2}$, $\mathbf{f}_\lambda^{n+1/2}$)
$t = t^n + h/2 \quad (n \leftarrow n + 1/2)$
Call CINT ($\mathbf{p}^{n+1/2}$, $\mathbf{g}^{n+1/2}$, h, $\mathbf{p}^{n+3/2}$)
Call LINT (ℓ^{n+1}, h, $\boldsymbol{\lambda}^{n+1}$, $\mathbf{f}_\lambda^{n+1}$)
$t = t^n + h$

Note that

$$\mathbf{g}^{n+1/2} = (\boldsymbol{\omega}^{n+1/2},\ f_\lambda^{n+1/2})$$

together with

$$\mathbf{p}^{n+1/2} = (\dot{\mathbf{d}}^n,\ \mathbf{d}^{n+1/2},\ \boldsymbol{\omega}^n,\ \mathbf{q}^{n+1/2})$$

provides the necessary input data to compute $\ddot{\mathbf{d}}^{n+1/2}$ and $\dot{\boldsymbol{\omega}}^{n+1/2}$ in the second call of CINT in the above calling sequence. In summary, the present procedure requires two function evaluations and two λ–solutions per each full step, hence the name "two-stage explicit-implicit staggered procedure".

VI. Numerical Examples

The two modules, the generalized coordinate integrator (CINT) and the Lagrange multipliers solver (LINT), have been implemented in Fortran 77. In solving the following three example problems, we have incorporated the constraint conditions through the use of Lagrange multipliers instead of eliminating the constraints. It is therefore necessary to solve the governing equations of motion in a way that satisfies the constraint equations. Hence, efficient and accurate solutions of these problems will confirm not only the viability of the present integration procedure for the solution of the multibody equations of motion with or without constraints but also the constraint stabilization procedure in their combined totality.

A. Plane Three-Link Manipulator

The first problem tested is a simplified version of the seven-link manipulator deployment problem[52]. The three links are initially folded and, for modeling simplicity, between the two joints is a coil spring which resists a constant deploying force at the tip of the third link. Also, the left-hand end of the first link is fixed through the same coil spring to the wall. These three coil springs are to be *locked up* once the links are deployed straight. The deployment sequence of the manipulator is illustrated in Fig. 1. The time-discretized difference equations both for Baumgarte's technique and the staggered stabilization technique have been solved at each time increment by a Newton-type iterative procedure to meet

a specified accuracy level. Hence, the performance of the two techniques can be assessed by the average number of iterations taken per time increment. This is presented in Fig. 2 for the accuracy of 10^{-4}. Notice that the staggered stabilization technique requires on the average about 4.5 iterations per step, whereas Baumgarte's technique requires about 22 iterations per step.

Note that Baumgarte's technique fails to converge for time, $t \approx 1.1$ as manifested in Fig. 2 because the rows in $\mathbf{B}$ become numerically dependent upon one another when the links are in a straight configuration. This corroborates the theoretical prediction of non-convergence whenever the solution matrix, $\mathbf{B}\mathbf{M}^{-1}\mathbf{B}^T$, for Baumgarte's technique (see Eqs.(5b), (20) and (21)) becomes singular. On the other hand, the staggered stabilization technique still converges within 30 iterations, because it overcomes this singularity difficulty, since $\dot{\boldsymbol{\lambda}}$ still exists, as can be seen from Eqs. (27) and (32).

It should be noted that, in order to avoid such ill-conditioning, one must differentiate the constraint equations once or twice more and recast the resulting higher-order constraint equations in terms of first-order equations with increased number of equations. This process is known as an index reduction strategy[46]. Thus, one must restructure the augmented equations of motion (5) with the net result of increased solution variables. Other techniques involve singular value decompositions, e.g., as advocated by Führer and Leimkuhler[53]. On the other hand, the present staggered stabilization technique overcomes the ill-conditioning difficulty without restructuring the governing equations of motion. Instead, the constraint equations are enforced in a separate module by the parabolically regularized equations for the Lagrange multipliers as derived in (27) and (32).

Although not reported here, the same relative performance has been observed for different accuracy levels, i.e., for the accuracy of 10^{-5} and 10^{-6}.

From this test problem, we conclude that the staggered stabilization technique yields both improved accuracy over and greater computational robustness than the Baumgarte technique. In addition, the staggered stabilization technique offers software modularity in that the solution of the constraint force, λ, can be carried out separately from that of the generalized displacement, $\mathbf{q}$. The only data each solution module needs to exchange with the other is a set of vectors, plus a common module to generate the gradient matrix of the constraints, $\mathbf{B}$. However, one should be cautioned not to extrapolate blindly to complex problems the results of the present simple examples. Further judicious experiments are needed in applying the present staggered stabilization technique to complex production-level problems before it can be adopted for general applications in multibody dynamic simulations.

B. Three-Dimensional Double Pendulum

The second problem with which we have tested the present procedure is a spatially moving double pendulum as shown in Fig. 3. The governing equations of motion become those of two separate rigid bars, except they are connected by two spherical joints. From Fig. 3

we have the the following quantities:

$$\Phi^i = \dot{\mathbf{d}}^i - \frac{1}{2}\boldsymbol{\omega}^i \times \mathbf{z}^i = 0, \quad i = 1,\ 2. \tag{57}$$

$$\mathbf{M} = \mathrm{diag}\{\mathbf{m}^1,\ \mathbf{J}^1,\ \mathbf{m}^2,\ \mathbf{J}^2\} \tag{58}$$

$$\mathbf{B} = \begin{bmatrix} \mathbf{I} & \frac{1}{2}\tilde{\mathbf{z}}^1\times & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & -\frac{1}{2}\tilde{\mathbf{z}}^1\times & -\mathbf{I} & -\frac{1}{2}\tilde{\mathbf{z}}^2\times \end{bmatrix} \tag{59}$$

$$\mathbf{F}\boldsymbol{\omega} = \begin{Bmatrix} f^1 \\ f^2 \end{Bmatrix}, \qquad f^i = -\begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega_2\omega_3(J_2 - J_3) \\ \omega_3\omega_1(J_3 - J_1) \\ \omega_1\omega_2(J_1 - J_2) \end{bmatrix}^i, \quad i = 1,\ 2. \tag{60}$$

$$\ddot{u}^i = \{\,\ddot{d},\ \dot{\omega}\,\}^i, \qquad \ddot{d}^i = [\ddot{x},\ \ddot{y},\ \ddot{z}]^T, \qquad \dot{\omega}^i = [\dot{\omega}_1,\ \dot{\omega}_2,\ \dot{\omega}_3]^T \tag{61}$$

$$\lambda = [\lambda_1,\ \lambda_2,\ \lambda_3,\ \lambda_4,\ \lambda_5,\ \lambda_6]^T \tag{62}$$

In the preceding equations, $\frac{1}{2}\mathbf{z}$ is the vectorial distance from the center of the bar to the spherical joint constraints, $\mathbf{m}$ and $\mathbf{J}$ are the three translational and rotatory inertia matrices, $\tilde{\mathbf{z}}$ is the skew symmetric matrix formed by the three components of $\mathbf{z}$, $\times$ implies a vector cross multiplication, and the superscript designates the i-th bar.

The pendulum is originally positioned in a gravity field with initial horizontal angular velocities ($\omega_z^{(1)} = \omega_z^{(2)} = 1$). Figure 4 shows the spatial trajectories of the two mass centers as projected on the horizontal surface and on the vertical plane. It is noted that the two trajectories form a similar pattern. The constraint forces and angular velocities, although not reported herein, exhibit patterns that are analogous in their characteristics for the two joints and two mass centers, respectively.

We have performed convergence studies by using different stepsizes h. Numerical evaluations indicate, as with the rigid-link problem, that when the stepsize samples more than 20 per period, the present procedure yields both good accuracy and stability.

C. Open-Loop Torque for Three-Link Manipulator

The third problem is a three-link manipulator maneuvering under a specified nonholonomic tip velocity constraint. For this problem, both rigid links and flexible links with four beam elements per link have been investigated. The flexible beam was modeled with a constant-strain Timoshenko beam element that allows large rotations. The three joints are modeled as spherical ones and the Lagrange multipliers have been introduced to enforce

the joint constraints and well as the nonholonomic constraint at the manipulator tip. The trajectories of the manipulator and the tip velocity specification are shown in Figs. 5 and 6. The corresponding joint torques for the rigid and flexible links are also shown in Figs. 7 and 8, respectively. Note that even though there exists little difference in the two trajectories of the rigid and flexible cases, there are significant differences in the open-loop joint torques. These will play an important role in the design of controller for vibration suppression in the manipulator arms.

VII. Discussions

In this chapter, we have presented a computational procedure for direct integration of the multibody dynamical (MBD) equations with constraints.

Because of its step-advancing nature, the procedure is labeled as a two-stage staggered explicit-implicit algorithm: explicit for solving the generalized coordinates (CINT) and implicit for Lagrange multipliers to incorporate constraints (LINT). Our numerical experiments indicate that it is essential to enforce the linear dependency constraint condition on the Euler parameters at each integration step.

Numerical experiments reported herein and additional applications conducted so far indicate that the present procedure yields robust solutions if the stepsize gives more than twenty samples for the period of the apparent highest response frequency of a given multibody system. Hence, the present procedure appears to have accomplished the following:

- For closed loop multibody systems and/or problems with complex topology wherein it is practically inadvisable to eliminate the constraints, the present procedure facilitates a straightforward construction of the governing equations of motion with appropriate constraints. The generalized coordinates and the system open and closed loop Lagrange multipliers can then be solved by the present procedure in a partitioned manner.
- For problems that involve lock-up mechanisms or similar discontinuities, the present procedure appears to overcome numerical difficulties encountered in using the Baumgarte stabilization. This may be an important impetus for applying the present procedure for the simulation of deployment dynamics of space structures.
- The angular velocity is obtained by an adaptation of the central difference algorithm in a two-stage form and the update of angular orientations is based on the Euler parameters by adopting the mid-point implicit formula. Both of the integrators conserve the system energy, which is important when the multibody simulation package is to be interfaced with an active control synthesis module. This is because stability margins of active control systems are sensitive to the system damping characteristics either physical or numerical.
- The present MBD solution procedure is implemented into two separate modules: the generalized coordinates solver (CINT) and the constraint Lagrange multiplier solver (LINT). Hence, the task for interfacing of the present MBD solution modules with additional capabilities such as active controller, observer and other analysis and design software modules becomes relatively straightforward. Such software architecture is

in contrast to most of the existing programming practice wherein several analysis capabilities are embedded into a single monolithic program.

Applications of the present procedure to flexible multibody systems are currently being carried out and preliminary results are quite encouraging. We hope to report on the results of flexible-body dynamics as well as on large-scale multibody problems in the near future.

Acknowledgements

The work reported herein was supported by NASA/Langley Research Center under Grant NAG-1-756. The authors wish to thank Dr. Jerry Housner for his keen interest and encouragement during the course of the present work.

References

1. Hooker, W. and Margulies, G., "The Dynamical Attitude Equations for an N-body Satellite," J. Astronautical Science, Vol. 12, 1965, pp. 123-12.

2. Roberson, R. and Wittenburg, J., "A Dynamical Formalism for an Arbitrary Number of Interconnected Rigid Bodies with Reference to the Problem of Satellite Attitude Control," Proc. the Third Int. Congress of Automatic Control, Butterworth, London, 1965.

3. Roberson, R., "A Form of the Translational Dynamical Equations for Relative Motion in Systems of Many Non-Rigid Bodies," Acta Mech. Vol. 14, 1972, pp. 297-308.

4. Huston, R. L. and Passerello, C. E., "On Lagrange's Form of d'Alembert's Principle," The Matrix and Tensor Quarterly, Vol. 23, 1973, pp. 109-112.

5. Boland, P., Samin, J. and Willems, P., "Stability Analysis of Interconnected Deformable Bodies in a Topological Tree," AIAA J., Vol. 12, 1974, pp. 1025-1030.

6. Likins, P., "Analytical Dynamics and Nonrigid Spacecraft Simulation," Jet Propulsion Laboratory, Technical Report 32-1593, Pasadena, Ca., 1974.

7. De Veubeke, B. F., "The Dynamics of Flexible Bodies," Int. J. Engng. Sci., Vol. 14, 1976, pp. 895-913.

8. Jerkovsky, W., "The Transformation Operator Approach to Multisystem Dynamics, Part I: The General Approach," The Matrix and Tensor Quarterly, Vol. 27, 1976, pp. 48-59.

9. Ho, J. Y. L., "Direct Path Method for Flexible Multibody Spacecraft Dynamics," Journal of Spacecraft and Rockets, Vol. 14, No. 2, 1977, pp. 102-110.

10. Wittenburg, J., Dynamics of Systems of Rigid Bodies, B. G. Teubner, Stuttgart, 1977.

11. W. W. Armstrong, "Recursive Solution to the Equations of Motion of an N-Link Manipulator," Proc. 5th World Congress, Theory of Machines, Mechanisms, Vol. 2, 1979, pp. 1343-1346.

12. Hollerbach, J., M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," IEEE Trans on Systems, Man, and Cybernetics, SMC-10, 1980, pp. 730-736.

13. Kane, T. and Levinson, D., "Formulation of Equations of Motion for Complex Spacecraft," J. Guidance and Control, Vol. 3, 1980, pp. 99-112.

14. Keat, J. E., "Dynamical Equations of Body Systems with Applications Space Structure Deployment", PhD Thesis, MIT, 1983.

15. Haug, E. J. (ed.), Computer Aided Analysis and Optimization of Mechanical System Dynamics, Springer-Verlag, Berlin 1984.

16. Roberson, R. E. and Schwertassek, R., Dynamics of Multibody Systems, Springer-Verlag, New York, 1984.

17. Schwertassek, R. and Roberson, R. E., "A State-Space Dynamical Representation for Multibody Mechanical Systems, Part II," Acta Mechanica, Vol. 51, 1984, pp. 15-29.

18. Bianchi, G. and Schielen, W. (eds), Dynamics and Multibody Systems, Springer-Verlag, Berlin, Heidelberg, 1986.

19. Huston, R. L., Lecture Notes on Dynamics, Preprint, University of Cincinnati, 1988.

20. Walton, W. C. and Steeves, E. C., "A New Matrix Theorem and Its Application for Establishing Independent Coordinates for Complex Dynamical Systems with Constraints," NASA TR-R326, 1969.

21. Gear, C. W., "Simultaneous Numerical Solution of Differential/Alge-braic Equations," IEEE Trans. Circuit Theory, CT-18, 1971, pp. 89-95.

22. Baumgarte, J. W., "Stabilization of Constraints and Integrals of Motion in Dynamical Systems," Comp. Meth. Appl. Mech. Engr., Vol. 1, 1972, pp. 1-16.

23. Orlandea, N., Chase, M. A. and Calahan, D. A., "A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems - Part I and II," Trans. ASME, J. Eng. for Industry, Ser. B, Vol. 99, 1977, pp. 773-784.

24. Lötstedt, P., "On a Penalty Function Method for the Simulation of Mechanical Systems Subject to Constraints," Royal Institute of Technology, TRITA-NA-7919, Stockholm, Sweden, 1979.

25. Orin, D. E., D. E., McGhee, R. B., Vukobratovic, M. and Hartoch, G., "Kinematic and Kinetic Analysis of Open-chain Linkages Utilizing Newton-Euler Methods," Math. Biosc., Vol. 43, 1979, pp. 106-130.

26. Huston, R. L. and Kamman, J. W., "A Discussion on Constraint Equations in Multibody Dynamics," Mech. Res. Comm., Vol. 9, 1982, pp. 251-256.

27. Petzold, L., "Differential/Algebraic Equations are not ODEs," SIAM J. Sci. Stat. Comp., Vol. 3, 1982, pp. 367-384.

28. Wehage, R. A. and Haug, E. J., "Generalized Coordinate Partitioning for Dimension Reduction in Analysis of Constrained Dynamic Systems," ASME J. of Mech. Design, Vol. 104, 1982, pp. 247-255.

29. Baumgarte, J. W., "A New Method of Stabilization for Holonomic Constraints," Journal of Applied Mechanics, Vol. 50, 1983, pp. 869-870.

30. Führer, C. and Wallrapp, O., "A Computer-Oriented Method for Reducing Linearized Multibody System Equations by Incorporating Constraints," Comp. Meth. Appl. Mech. Engr., Vol. 46, 1984, pp. 169-175.

31. Nikravesh, P. E., "Some Methods for Dynamic Analysis of Constrained Mechanical Systems: a Survey," in: Computer Aided Analysis and Optimization of Mechanical System Dynamics (E. J. Haug, ed.), NATO ASI series, F9, Springer-Verlag, Berlin, 1984, pp. 351-367.

32. Enright, W. H., "Numerical Methods for Systems of Initial Value Problems - The State of the Art," in: Computer Aided Analysis and Optimization of Mechanical System Dynamics(E. J. Haug, ed.), NATO ASI series, F9, Springer-Verlag, Berlin, 1984, pp. 309-322.

33. Garcia de Jalon, J., Unda, J., Avello, A. and Jimenez, J. M., "Dynamic Analysis of Three-Dimensional Mechanisms in Natural Coordinates," Journal of Mechanisms, Transmissions and Automation in Design, Vol. 109, 1987, pp. 460-465.

34. Park, K.C., Chiou, J.C. and J. D. Downer, "A Computational Procedure for Large Rotational Motions in Multibody Dynamics ," Proc. the 29th Structures, Dynamics and Materials Conference, AIAA Paper No. 88-2416, AIAA, 1988, pp.1593-1601 (also to appear in J. Guidance, Control and Dynamics).

35. Park, K. C. and Chiou, J. C., "Stabilization of Computational Procedures for Constrained Dynamical Systems," Journal of Guidance, Control and Dynamics, Vol. 11, July-August 1988, pp. 365-370.

36. Geradin, M. and Cardona, A., "Kinematic and Dynamics of Rigid and Flexible Mechanisms Using Finite Elements and Quaternion Algebra," Computational Mechanics, Vol. 4, 1989, pp. 115-135.

37. Bodley, C. S., Devers, A. D., Park, A. C. and Frish, H. P., "A Digital Computer Program for the Dynamic Interaction Simulation of Controls and Structures (DISCOS)," NASA Technical Paper 1219, 1978.

38. The ADAMS User's Guide, Mechanical Dynamics, Inc., Ann Arbor, Mich., 1979.

39. Schwertassek, R., "Der Roberson/Wittenburg Formalismus and das Programmsystem MULTIBODY zur Rechnersimulation von Mehrköpersystemen," Report DFVLR-FB-78/08, DFVLR, Köln, 1978.

40. Huston, R. L., Harlow, M. W. and Gausewitz, N. L., "User's Mannual for UCIN-EULER - A Multipurpose, Multibody Systems Dynamics Computer Program," NTIS Report AD-A120403, 1982.

41. Haug, E. J., Lance, G. M., Nikravesh, P. E., Vanderploeg, M. J. and Wehage, R. A., DADS (Dynamic Analysis and Design Systems), Computer Aided Design Software Inc., Oakdale, Iowa, 1985.

42. Housner, J. M., McGowan, P. E., Abrahamson, A. L. and Powel, M. G., "The LAT-DYN User's Mannual," NASA TM 87635, NASA/Langley Research Center, January 1986.

43. Hughes, T. J. R. and Belytschko, T., "A Prècis of Developments in Computational Methods for Transient Analysis," Journal of Applied Mechanics, **50**, 1983, 1033-1041.

44. Park, K. C., "Transient Analysis Methods in Computational Methods," *Finite Elements: Theory and Applications* (ed. D. L. Dwoyer, M. Y. Hussaini and R. G. Voigt), Springer-Verlag, 1988, 240-267.

45. Belytschko, T., Englemann, B. E. and Liu, W. K., "A Review of Recent Developments in Time Integration," in: State-of-the-Art Surveys on Computational Mechanics (Noor, A. K. and Oden, J. T., editors), ASME, 1989, pp. 185-200.

46. Brenan, K. E., Campbell, S. L. and Petzold, L. R., *The Numerical Solution of Initial Value Problems in Ordinary Differential-Algebraic Equations*, Elsevier Science Publishing Co., 1989.

47. Park, K. C., "Partitioned Analysis Procedures for Coupled-Field Problems: Stability Analysis," Journal of Applied Mechanics, Vol. 47, 1980, pp. 370-378.

48. Felippa, C. A. and Park, K. C., 'Staggered Transient Analysis Procedures for Coupled Mechanical Systems," Computer Methods in Applied Mechanics and Engineering, Vol. 24, 1980, pp. 61-111.

49. Park, K. C. and Felippa, C. A., "Partitioned Analysis of Coupled Systems," in Computational Methods for Transient Analysis, T. Belytschko and T. J. R. Hughes (eds.), Elsevier Pub. Co., 1983, pp. 157-219.

50. Felippa, C. A. and Park, K. C., "Computational Aspects of Time Integration Procedures in Structural Dynamics, Part 1: Implementation," Journal of Applied Mechanics, Vol. 45, 1978, pp. 595-602.

51. Lanczos, L., The Variational Principles of Mechanics, 4th ed., University of Toronto Press, 1970, pp. 141-147.

52. Housner, J. M., "Convected Transient Analysis for Large Space Structure Maneuver and Deployment," AIAA-84-1023-CP, *Proc. 25th Structures, Structural Dynamics and Material Conference*, Part 2, 14-16 May 1984, Palm Springs, pp. 616-619.

53. Führer, C. and Leimkuhler, B., "Formulation and Numerical Solution of the Equations of Constrained Mechanical Motion," Technical Report DFVLR-FB 89-08, DFVLR, D-5000 Köln 90, 1989.

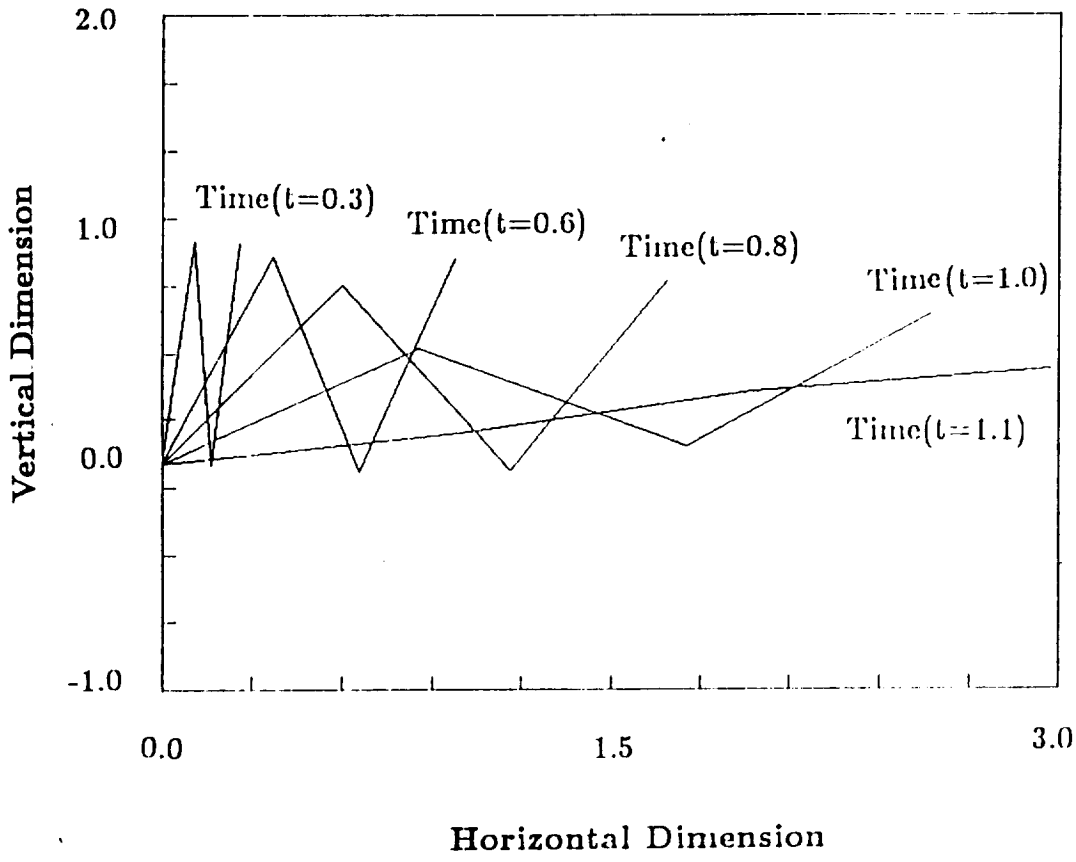
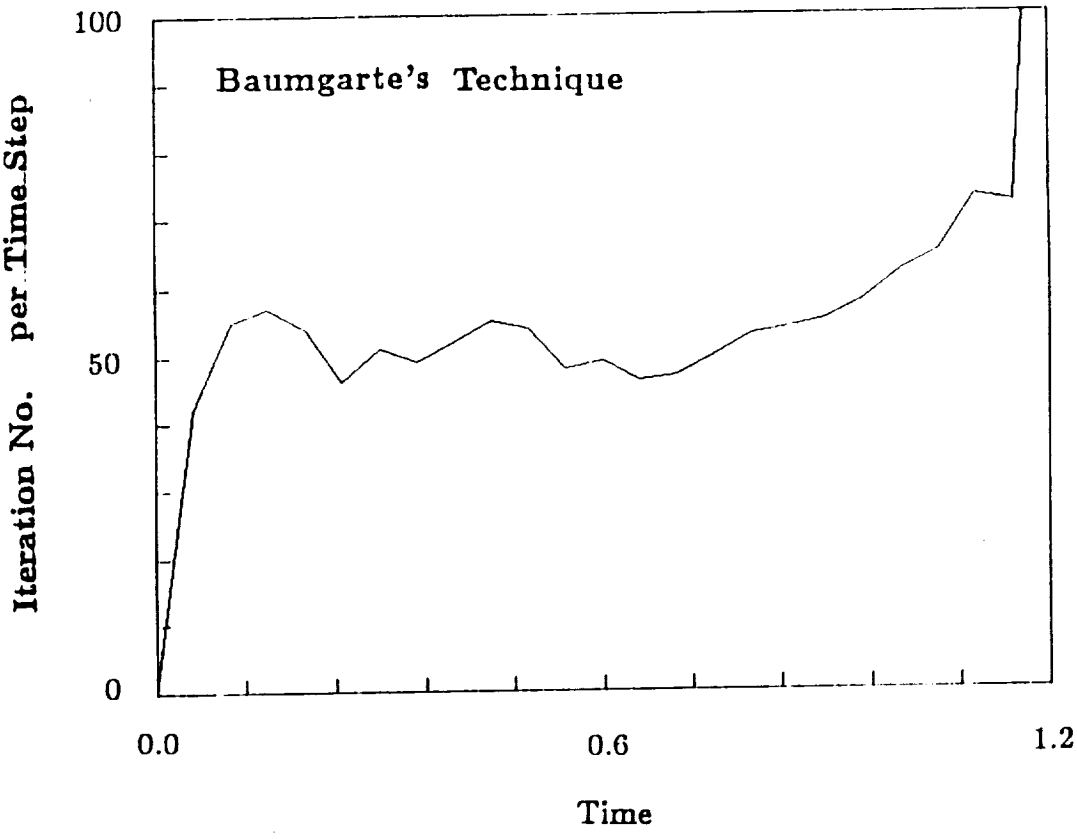


Fig. 1 Deployment of Three-Link Remote Manipulator



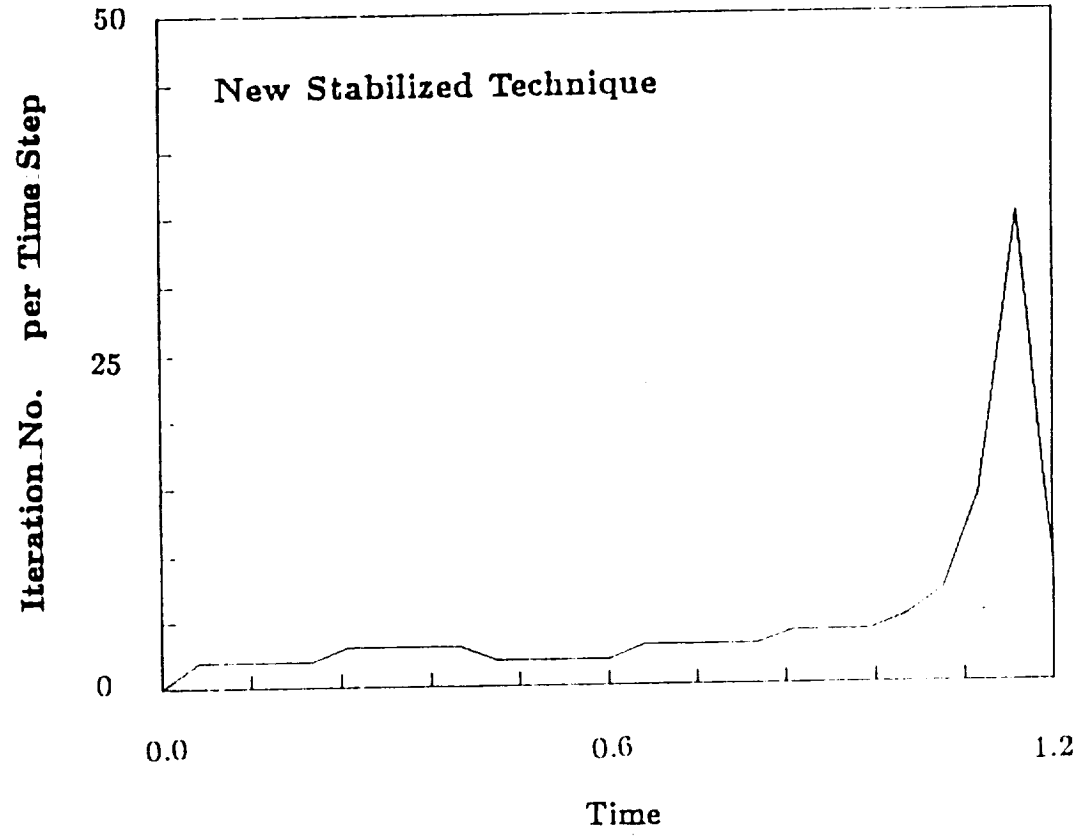


Fig 2 Performance of Two Stabilization Techniques for Three-Link Remote Manipulator (Solution Accuracy=10^{-6})

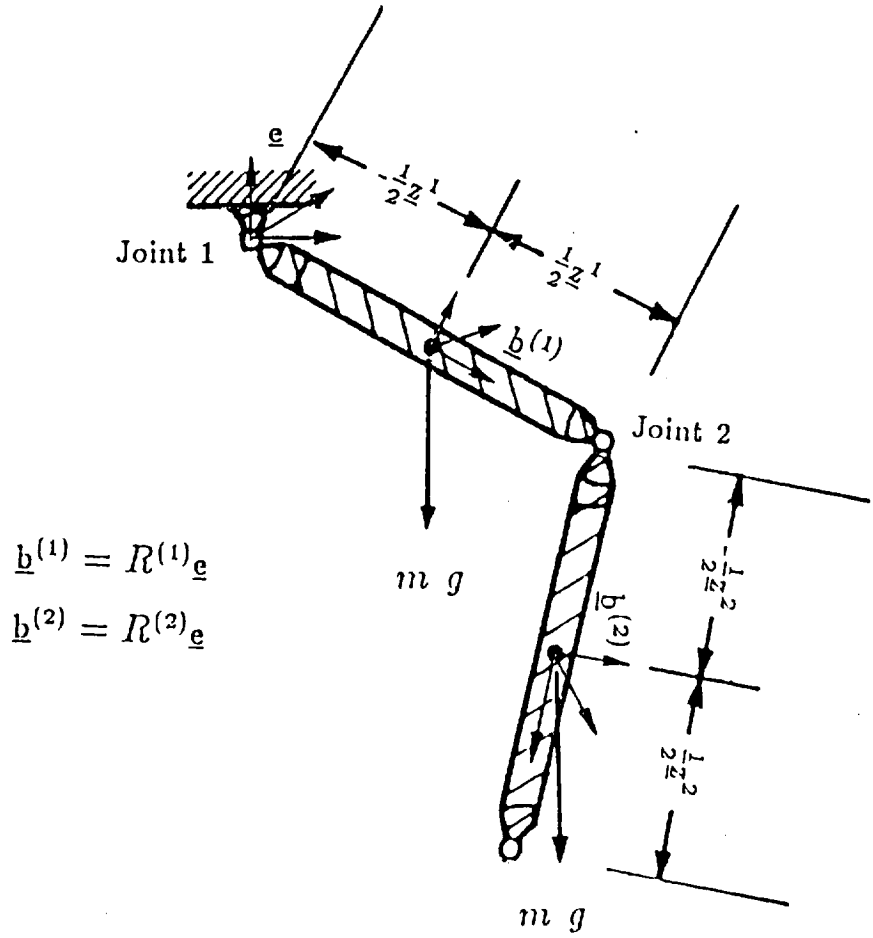


Fig 3 Double Pendulum with Spatial Joints

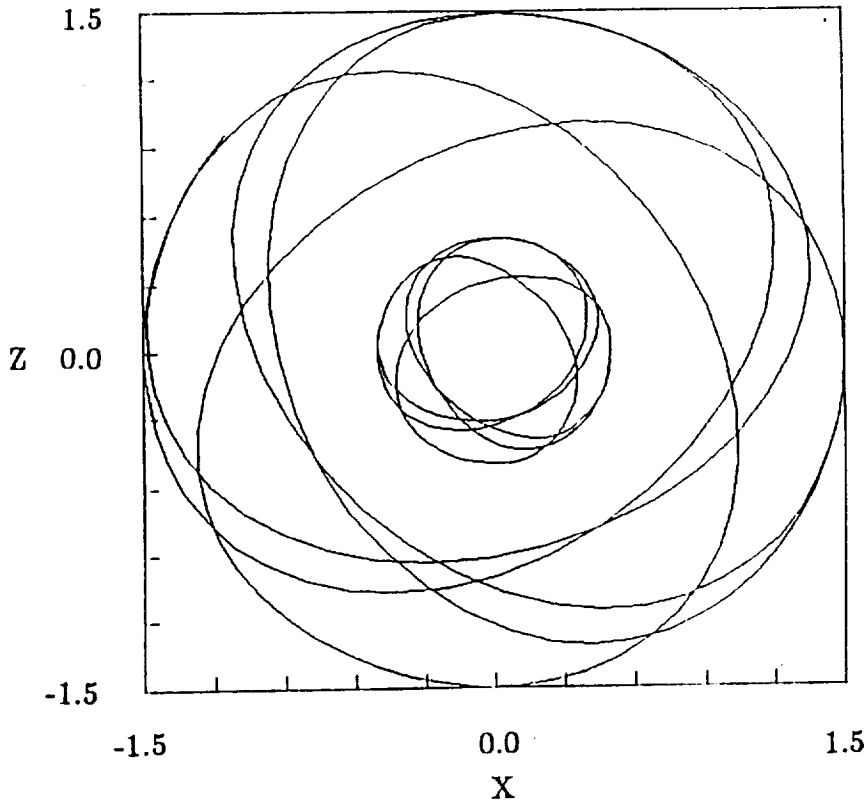


Fig. 4a Trajectories of double pendulum on X-Z plane

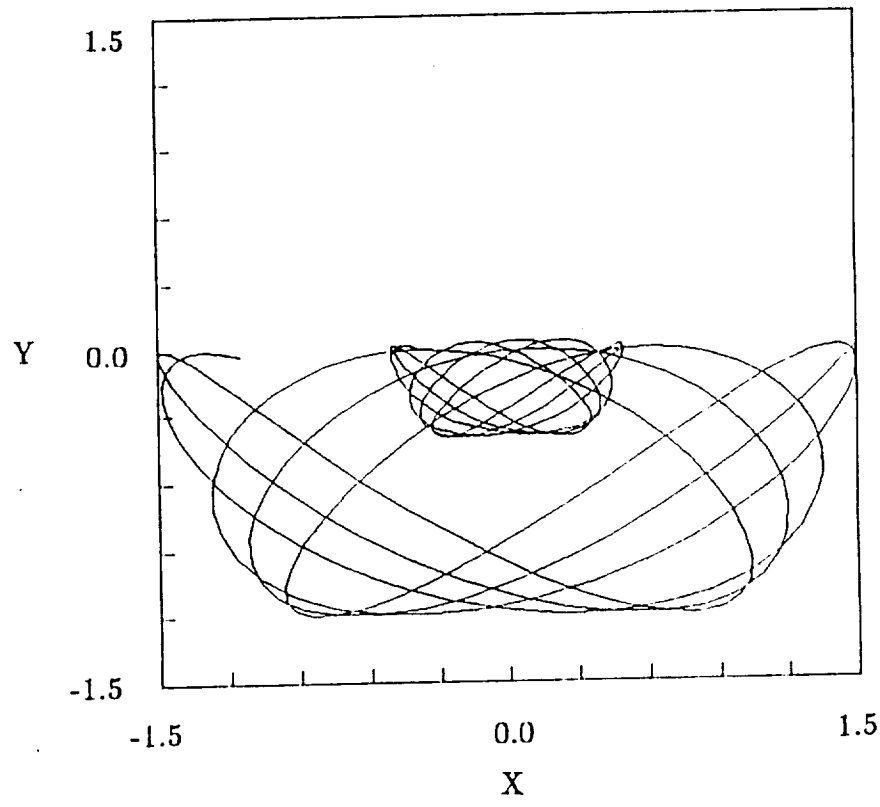


Fig. 4b Trajectories of double pendulum on X-Y plane

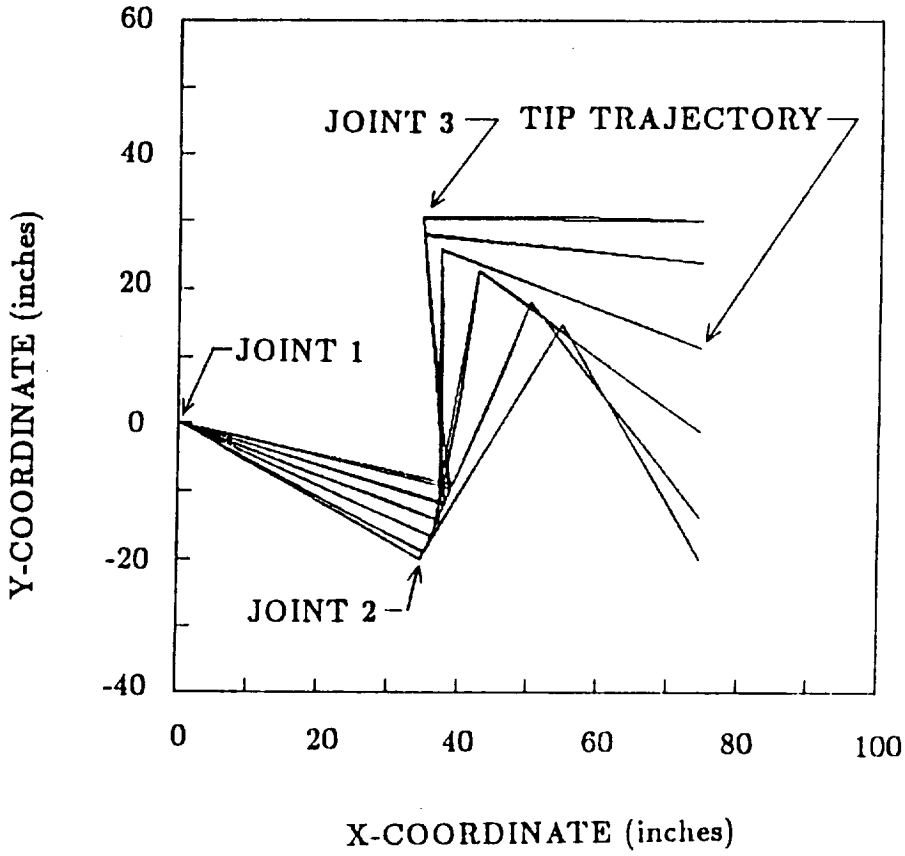


Fig. 5 Crane Tip Trajectory of Rigid and Flexible Members

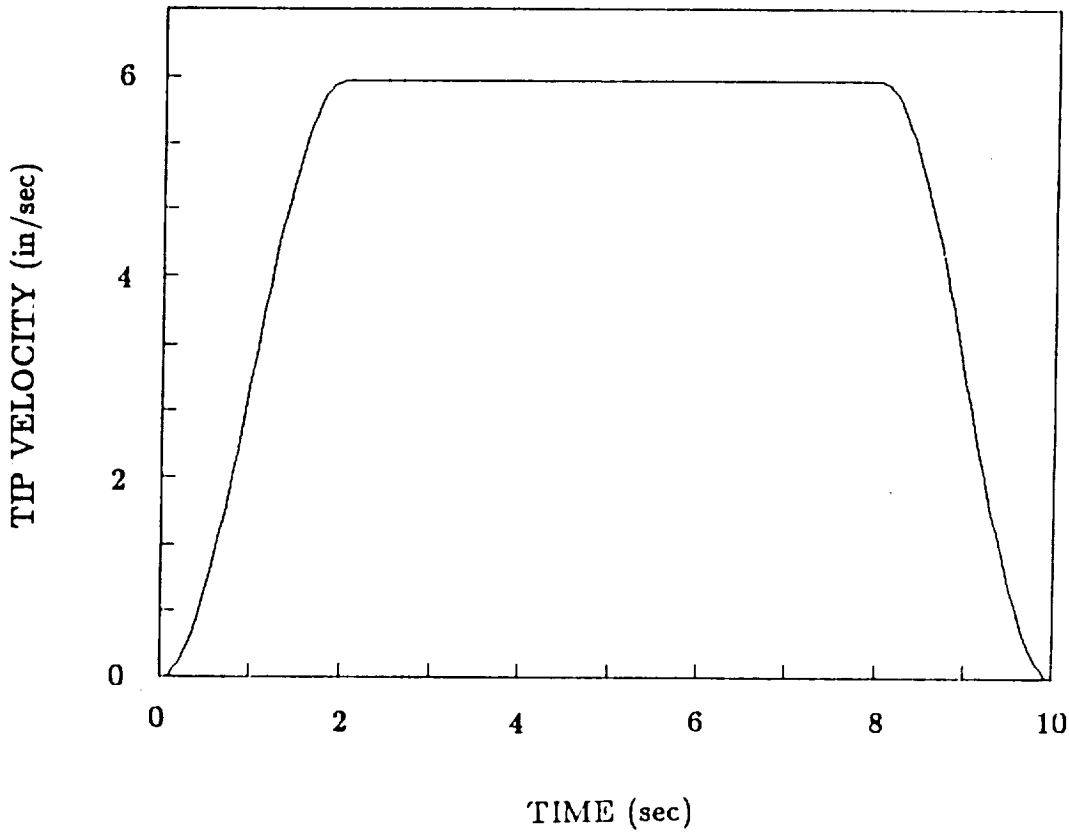


Fig. 6 Crane Tip Velocity of Rigid and Flexible Members

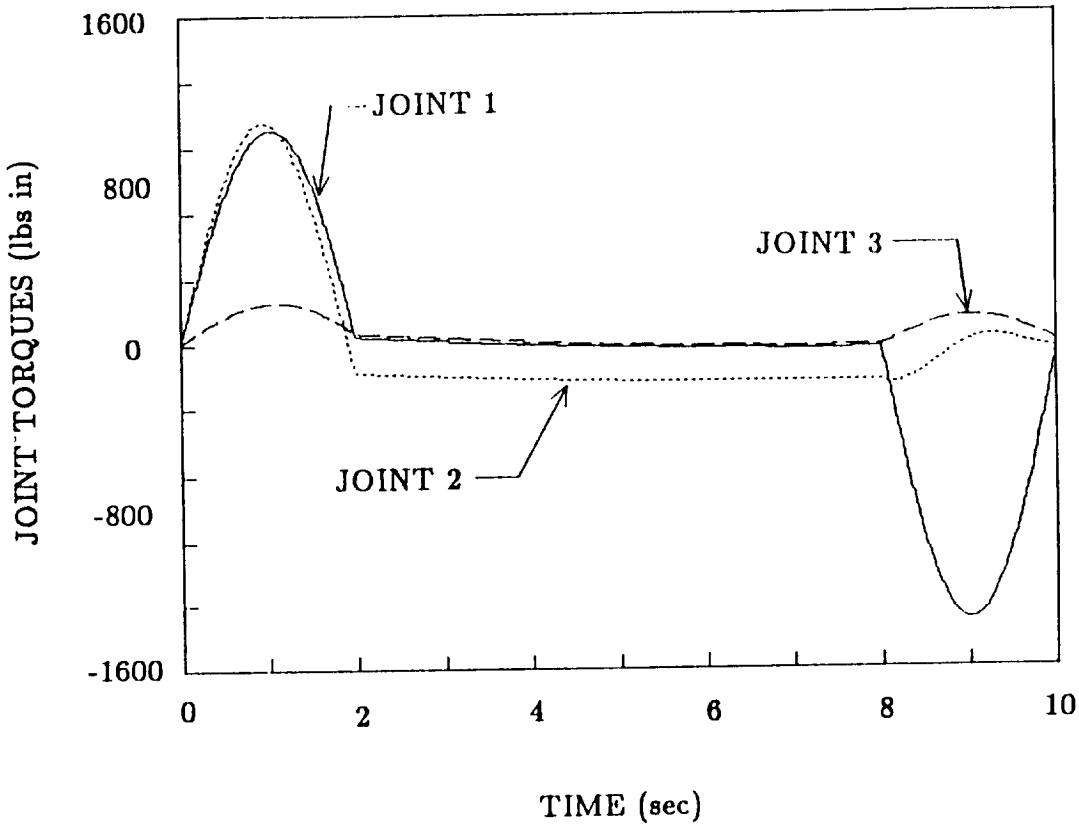


Fig. 7 Crane Joint Torque (Rigid Members) vs. Time

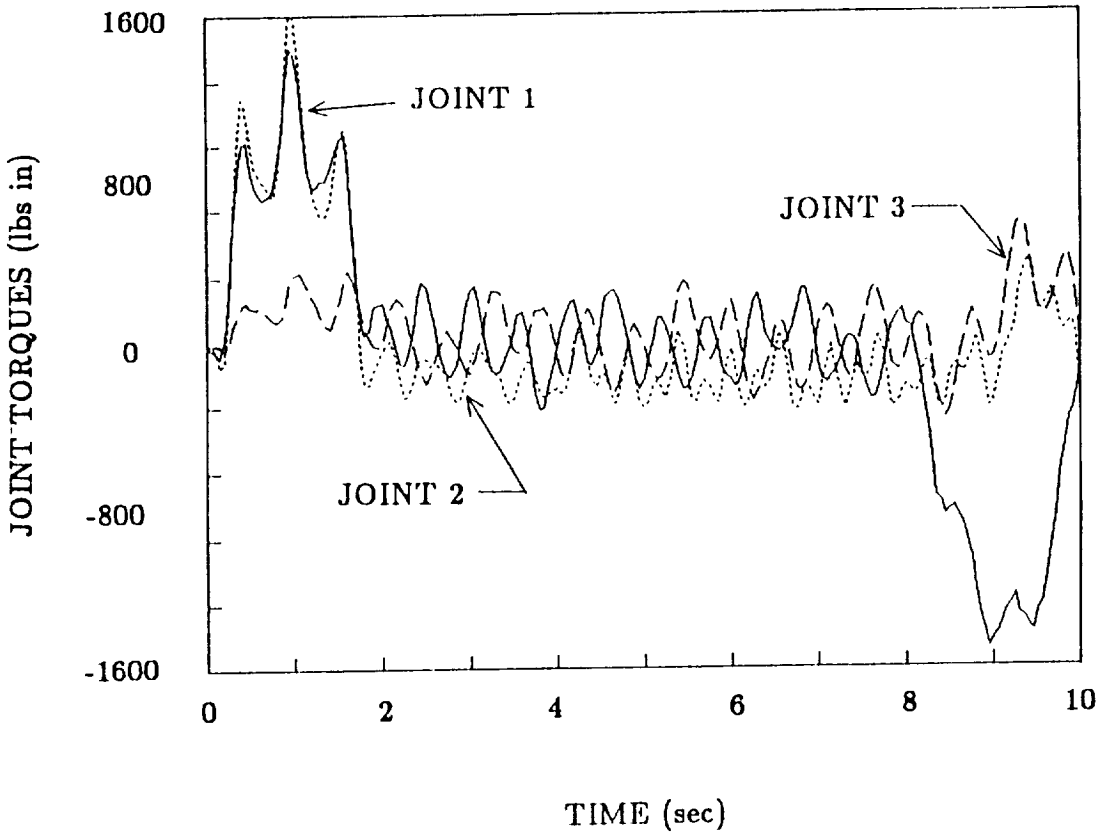


Fig. 8 Crane Joint Torque (Flexible Members) vs. Time